

Week 2: Assignment

1. Which of the following statements is false for Verilog modules?
 - a. A module cannot contain definitions of other modules.
 - b. When a module X is called multiple numbers of times from some other module, only one copy of module X is included in the hardware after synthesis.
 - c. More than one module cannot be instantiated within another module.
 - d. If a module X is instantiated 4 times within another module, 4 copies of X are created.

Answer: (b) and (c)

Option (a) is true as verilog modules declarations must be disjoint. When a module X called N number of times from another module, N copies of X are embedded inside the module. Thus option (b) is false and option (d) is true. Any number of modules can be invoked from within a module. Thus option (c) is false.

2. What does the statement the verilog statement “assign f = !(a & b) | !(a ^ b)” signify?
 - a. The value at LHS gets changed whenever there is a change in the expression at RHS.
 - b. The LHS is a register type variable.
 - c. The variables at RHS must be wire type.
 - d. Specification of a combinational circuit design at behavioral level.

Answer: (a) and (d)

An assign statement represent continuous assignment, where the variable on the LHS gets updated whenever the expression on the RHS changes. Thus option (a) is correct. The LHS of an assign statement must be net type whereas RHS can contain both net type as well as register type variables. Thus option (b) and (c) are incorrect. The assign statement is typically used to model combinational circuit at the behavioral level.

3. Which of the following is not true for net type variables?
 - a. It represents connections continuously driven by the outputs of the connected device.
 - b. It can be used in an expression on the RHS of an “assign” statement.
 - c. Once a value is assigned, it will hold the value.
 - d. None of the above.

Answer: (c)

A net type variable is continuously driven by the output of a gate or module. Thus option (a) is true. A net type variable may appear on the RHS of an assign statement. Thus option (b) is also true. Since net type variable is continuously driven, the value gets changed whenever the output of the connected gate or module change. Thus option (c) is false.

4. For the following Verilog code segment, what will be the number of bits in S as deduced during synthesis?

```

wire [31:0] X, Y;
integer S;
S = X + Y + 1;

```

- a. 62
- b. 32
- c. 33
- d. None of the above

Answer: (c)

Since X and Y are of 32 bits size, they represent an unsigned number as large as $2^{32} - 1$. Adding such large numbers like $X + Y + 1 = 2^{32} - 1 + 2^{32} - 1 + 1 = 2^{33} - 1$, i.e. S will be of size 33 bits. Thus option (c) is the right answer.

5. For the following Verilog code segment, if the initial value of IR is 3456ABCD (in hexadecimal), the value of “data” in decimal will be (Note that “data” is a 4-bit variable)

```

wire [31:0] IR;
wire [3:0] data;
wire [15:0] d1;
wire [31:16] d2;
assign d1 = IR[31:16];
assign d2 = IR[15:0];
assign data = d1[7:4] + d2[23:20] + d2[27:24];

```

Answer: 12

The assignment yields:

```

d1    = 3 4 5 6 (hexadecimal)
d2    = A B C D (hexadecimal)
data  = 5 + C + B (hexadecimal)
      = 0101 + 1100 + 1011 (binary)
      = 1100 (with a carry 1)
      = 12

```

6. Consider the following Verilog module.

```

module guess (data, cond, result);
    input [7:0] data;
    input [1:0] cond;

```

```

output reg result;
always @(data)
begin
    if (cond == 2'b00) result = |data;
else result = ~^data;
end
endmodule

```

Which of the following are true when the module is synthesized?

- a. A combinational circuit will be generated.
- b. A sequential circuit with a storage element for result will be generated.
- c. The synthesizer system will generate a wire for result.
- d. None of the above.

Answer: (a) and (c)

Since values assigned to variable result for all possible values of “cond” variable, no latch will be generated and the synthesized circuit will be combinational with wire type result output. Thus option (a) and (c) are true.

7. For the following Verilog code segment:

```

wire [7:0] A;
wire B;
assign B = ~|A;

```

if the value of A is 8'b00111001, what will be the value of {A[5:3], 3{B}}?

- a. 6'b111111
- b. 6'b011000
- c. 6'b111000
- d. None of the above

Answer: (c)

The bitwise NOR operation on A will result in 0, i.e B = 0 where A = 8'b00111001. Thus the concatenation of A[5:3] (=111) with 3 times replication of B will result in 111000. Thus option (c) is true.

8. Which of the followings is true regarding \$monitor statement used in Verilog test bench?

- a. It used to print the values of specified variables.
- b. The values are printed whenever changes detected in any of the specified variables.
- c. The printing occurs at the place \$monitor statement is encountered.
- d. None of the above.

Answer: (a) and (b)

The \$monitor statement is used to print the values of specified variable whenever changes detected in any of these variables. Thus both option (a) and (b) is true. Option (c) is false as printing does not occur on encountering the \$monitor statement rather when one or more specified variables' state changes.

9. Which of the following statements is true for Verilog primitive gates?
- a. In a behavioral specification a variable number of logic gates can be instantiated inside a module.
 - b. The input and output ports of logic gates must be connected to wire type or register type variables.
 - c. The optional delay in gate instantiation is only used during simulation and ignored during logic synthesis.
 - d. All logic gates only respond to logic values 0 and 1.

Answer: (c)

Logic gates are instantiated in structural specification. Thus option (a) is false. The input ports of logic gates must be connected to wire type variables whereas output ports can be connected to wire type or register type variables. Thus option (b) is false. The delay in gate instantiation is used only for simulation. Thus option (c) is true. Logic gates respond to all possible logic values 0, 1, x, z. Thus option (d) is also false.

10. Identify the valid statements for Verilog operators.
- a. The bitwise operators like & and | can acts as unary as well as binary operator.
 - b. The LHS of "assign" operator can be register type or net type variable.
 - c. The shift operators X>>2 and X>>>2 yields similar results when X is a signed number.
 - d. The equality operator == and === returns same logic value when either of the operands contains all possible logic values.

Answer: (a)

The bitwise operators & and | performs bitwise AND and OR operation when operating on two operands whereas these operators acts as reduction operator when operating on single operand. Thus option (a) is true. The LHS of "assign" operator must be net type. Thus option (b) is false. The shift right operator >> does not extend the sign bit whereas the arithmetic right shift operator >>> extends the sign bit. Thus option (c) is false. The == operator compare only logic values 1 and 0 whereas === operator tests for all possible logic values 0, 1, z, and x. Thus option (d) is also false.